REMARKS

Claims 11-28 are pending in the present application. Claims 1-10 have been previously canceled. Applicants thank the examiner for consideration of the Information Disclosure Statements filed on January 31, 2005 and for acknowledging Applicants' claim for foreign priority under 35 U.S.C. § 119(a)-(d), and receipt of the certified copy entered into the application file.

Drawings

The drawings have been objected to for failing to comply with 37 CFR 1.84(p)(5), because the drawings are labeled with "FIG", but are referred to in the specification as "illustration". The specification has been amended to describe the drawings as they are labeled on the drawing sheets. No new matter has been added by this amendment. Reconsideration of this objection is requested.

Claim Rejections – 35 USC §101

Claims 11-28 were rejected under 35 U.S.C. § 101 for being directed to non-statutory subject matter. Independent claim 11, from which claims 12-24 depend and independent claim 27, from which claim 28 depends, have both been amended to recite that the software is encoded on a computer readable medium. Independent claim 25, from which claim 26 depends, has been amended to recite that the software description language is encoded on a computer readable medium. As amended, claims 11-28 do not recite a computer program, component, file or code per se. Rather, each recites software (or a software description language) as encoded on a computer readable

medium. Accordingly, Applicants suggest that the rejection has been overcome and request that it be withdrawn.

Claim Rejections under 35 USC § 102

Claims 11-18 were rejected under 35 U.S.C. § 102(e) as being anticipated by US 7,197,739 to Preston et al ("Preston"). Claim 11, from which claims 12-18 depend, recites "distributing said description between several code generators according to modifiable distribution rules." In contrast, Preston discloses a system with only a single code generator. Preston, Fig. 1b and col. 5, ln. 63 – col. 6, ln. 3. The system of Preston is precisely what the present specification defines as prior art; a code generator that can only generate code in one language at a time. Present specification, page 3. The system of Preston is unable to take advantage of generating code by using the advantages of several code generators, much less distributing the code generation task according to modifiable rules to optimize performance.

Basically, Preston teaches using a code generator for generating a processable computer program for a terminal unit based on a description in a natural language provided by the user. Preston, col. 5, ll. 63-67 and col. 6, ll. 20-24.

Therefore, Preston teaches the use of a data storage 106 in which predefined functions are stored. For each predetermined function, data storage 106 contains the corresponding computer code and an associated description of this function in natural language. Preston, col. 5, ll. 23-33.

Preston teaches using a data analyzer 102 for performing a semantic analysis of the description in natural language provided by the user in order to extract the functional requirement. Then, the data analyzer 102 looks for corresponding functions based on the associated function descriptions in data storage 106. When a corresponding function description is found in data storage 106, then data analyzer 102 retrieves the associated computer code in data storage 106. Based on the computer code of the retrieved functions, the code generator 103 generates the processable computer program. Preston, col. 5, l. 67 - col. 6, l. 20.

But Preston only makes use of one code generator.

Preston does not teach distributing the description (in natural language) entered by the user between <u>several</u> code generators and even less that such a distribution occurs according to <u>modifiable distribution rules</u>.

In other words, Preston does not disclose the that "the software generates the said computer code from a description of said at least one part of the computer application <u>by distributing said description between several code generators according to modifiable distribution rules</u>."

The Office cited Preston col. 5, l. 63 - col. 6, l. 3 and Fig. 1b in combination with Preston, col. 6, ll. 19-29.

But this is not correct:

-        Preston col. 5, l. 63 - col. 6, l. 3 only discloses the fact that the user provides a description in natural language of the computer code to be generated; and

-        Preston, col. 6, ll. 19-29 only discloses that the single code generator 103 achieves to translate this description in natural language into a processable computer program and loads it onto a terminal.

There are not several code generators involved here, but only one (code generator 103) and for this reason, there are no distribution rules according to which the description in natural language is distributed between several code generators.

In particular regarding Preston, col. 6, ll. 19-29, the Office cited using predetermined rules from the code generator 103 to create a processable computer program and distribute them on a terminal or network.

But these rules are those of code generator 103 alone; they do not correspond to distribution rules for distributing the description between several code generators as claimed. Further, as stated by the Office, these rules are predetermined while the claimed distribution rules are modifiable.

Furthermore, the Office writes "distribute them on a terminal or network," (where "them" seems to refer to the generated code) but this does not correspond to the claimed distribution of a description of a computer application between several code generators.

The invention does not relate to a code generator per se. On the contrary, one element of the claimed invention lies in distributing said description between several code generators according to modifiable distribution rules.

Preston, col. 7, II. 23-26 only mentions that Preston is not limited to the use of Java. But this does not suggest the fact of using several code generators in combination with distributing the description of a computer application between several code generators according to modifiable distribution rules as claimed.

The claimed invention allows one to alleviate the drawbacks of the prior art that are mentioned in p.3, II. 29-38. See also p.7, II. 27-32, p.8, II.1-2 and p.8, I.37 - p.9, I.3.

One advantage of the claimed invention is to allow an easy development of computer applications.

In the present invention, the user on one hand prepares a description of the computer application and on the other hand he defines distribution rules indicating how the constitutive elements of the description shall be distributed among different code generators which will generate code for the respective parts (which may each work with a respective language) of the technical architecture which will run the computer application. Schematically, the invention allows the user to define on the one hand a description of the computer application and on the other hand a description of the technical architecture (i.e. the claimed distribution rules) that will run the computer application. Based on this input, the software of the invention generates the computer code of the computer application for the different parts of the technical architecture in the appropriate languages.

This differs from the prior art mentioned in p.3, I.29-38 where the user has to self select which part of the description of the computer application to provide to each code generator (corresponding each to a part of the technical architecture).

Further, the invention allows an easy re-use of the computer application on another technical architecture. Indeed, the user just has to modify the description of the technical architecture (i.e. the distribution rules) in accordance with the new technical architecture without changing the description of the computer application. Based on

that, the software of the invention will generate the new computer code for the new technical architecture.

None of the cited prior art suggests a software generating computer code from a description of at least one part of the computer application by distributing said description between several code generators according to modifiable distribution rules, each code generator translating the part of said description that it is provided with, in order to provide at least one part of the said computer code in a respective language.

As a result, the invention of claim 1 is non-obvious over the prior art. Applicants suggest that this rejection has been overcome, and request that claims 11-18 be passed to issuance.

Claim Rejections – 35 USC § 103

Claims 19-24 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Preston in view of US 5,499,371 to Henninger ("Henninger"). Claims 19-24 depend from claim 11, and thus are allowable for at least the reasons above as Henninger does not cure the above defects of Preston. Applicants suggest that this rejection has been overcome, and request that claims 19-24 be passed to issuance.

Claim Rejections – 35 USC 102(b)

Claims 25-28 were rejected under 35 U.S.C. § 102(b) as being anticipated by Henninger. Claim 25 recites a software description language while Henninger does not deal with a software description language, but rather with what one could call a database description language.

Actually, according to what Henninger states in col. 3, ll. 13-22, in conjunction with col. 4 ll. 14-20, Henninger does not provide a means for building code for an application "using a set of object instances that work in cooperation to perform useful work" (emphasis added) but rather defines what method should be used to generate code that solely ties individual objects to information stored in a relational database by implementing four explicitly defined tasks limited to raw data manipulation : "create, retrieve, update and delete" and the said method does not disclose the recited object

cooperation to make an application from a collection of objects and thus perform useful work.

Henninger does not teach that some language, and in particular the object model, is organized in classes enabling one to define first classes giving access to technical and functional services to be provided by a hardware and software computer platform receiving the computer application, in which:

-        the said services cannot be defined by said language, and

-        the other classes cannot have access to any one of these technical or functional services except through said first classes.

Henninger, col. 9, ll. 37-50 and col. 9, l. 60 - col.10, l. 22 pointed at by the Office deal with key attributes of classes, but not with the ability of classes themselves with respect to other classes as claimed in claim 25.

Further, as stated in col. 3 ll. 13-22, the classes disclosed by Henninger all support the same four services to access data in a relational database. These four services are the only ones that are comprised in the disclosure of Henninger and all of these services are supported directly by all of the classes. One can easily deduce that there is no such notion in Henninger as being claimed in the present application, namely: providing specific services that cannot be defined within the language itself and are accessed only through the said first classes. Thus, the invention of claim 25 is neither anticipated nor suggested by Henninger. Applicants suggest that this rejection has been overcome, and request that claims 25 and 26 be passed to issuance.

Claim 27, from which claim 28 depends, recites the patentable elements of claim 25, and thus is allowable for at least the same reasons as above. Additionally, Henninger does not teach the software description language as recited in claim 25, so Henninger can even less teach software enabling to provide a description of a computer application model in such a software description language as claimed in claim 27.

The Office refers to the GUI of the computer system used by Henninger. The claimed invention is not merely a GUI, but software enabling to provide a description of a computer application model in a specific software description language which is defined in claim 27. Thus, the invention of claim 27 is neither anticipated nor suggested

by Henninger. Applicants suggest that this rejection has been overcome, and request that claims 27 and 28 be passed to issuance.

Conclusion

Accordingly, it is believed that all claims in this application are in allowable condition, and early and favorable reexamination is requested.

If any additional payment is required, please charge the cost thereof to deposit account no. 02-2135.

Respectfully submitted,

By: /David B. Orange/
David B. Orange
Attorney for Applicant
Registration No. 55,513
ROTHWELL, FIGG, ERNST & MANBECK
1425 K. Street, Suite 800
Washington, D.C. 20005
Telephone: (202) 783-6040

RBM/DBO:kdm

1523491_1